Инженерная школа информационных технологий и робототехники

Направление 09.03.01 «Информатика и вычислительная техника»

Отделение информационных технологий

Отчёт по теме

**«Автоматизированная система контроля посещаемости студентов»**

по дисциплине «Творческий проект»

Выполнили:

студент гр. 8В21 \_\_\_\_\_\_\_\_\_\_\_\_\_ Коршунов А.Е.

\_\_\_.\_\_\_.\_\_\_\_\_\_

студент гр. 8В21 \_\_\_\_\_\_\_\_\_\_\_\_\_ Востриков М.Н.

\_\_\_.\_\_\_.\_\_\_\_\_\_

студент гр. 8В21 \_\_\_\_\_\_\_\_\_\_\_\_\_ Тусупбеков Т.

\_\_\_.\_\_\_.\_\_\_\_\_\_

студент гр. 8В21 \_\_\_\_\_\_\_\_\_\_\_\_\_ Куц И.С.

\_\_\_.\_\_\_.\_\_\_\_\_\_

Проверил:

Директор отделения ИТ

оценка (до 30 баллов): \_\_\_\_\_\_\_\_\_\_\_\_\_

к защите допускаю \_\_\_\_\_\_\_\_\_\_\_\_\_ Фадеев А.С.

\_\_\_.\_\_\_.\_\_\_\_\_\_

**План:**

1. [ТЕХНИЧЕСКОЕ ЗАДАНИЕ ПРИЛОЖЕНИЯ](#_ТЕХНИЧЕСКОЕ ЗАДАНИЕ ПО ГОСТ 19.201-78)
2. [Целевая аудитория приложения](#_ЦЕЛЕВАЯ АУДИТОРИЯ:)
3. [Обзор аналогов](#_ОБЗОР АНАЛОГОВ:)
4. [Обзор интерфейса](#_ОБЗОР ИНТЕРФЕЙСА:)
5. [UML-диаграмма](#_UML-ДИАГРАММА ПРОЕКТА:)
6. [Реализация front-end части](#_РЕАЛИЗАЦИЯ FRONT-END ЧАСТИ)
7. [Реализация back-end части](#_РЕАЛИЗАЦИЯ BACK-END ЧАСТИ)
8. [Реализация работы с СУБД](#__x000F_РЕАЛИЗАЦИЯ РАБОТЫ С СУБД)
9. [Выводы, планы на следующий семестр](#_ВЫВОДЫ, ПЛАНЫ НА СЛЕДУЮЩИЙ СЕМЕСТР)
10. [Источники информации](#_ИСТОЧНИКИ ИНФОРМАЦИИ)
11. [Приложения](#_ПРИЛОЖЕНИЕ 1.)

# Техническое задание по ГОСТ 19.201-78

## **1.1. Наименование программы**

Наименование программы - «Автоматизированная система контроля посещаемости студентов (далее АСКПС)»

## **1.2. Краткая характеристика области применения**

Система «АСКПС» предназначена для облегчения и повышения качества учёта посещаемости студентов. В течение рабочего дня (8:30-21:55) будет исполняться комплекс программ, производящий автоматический сбор информации о присутствии студентов (очно) на паре. По окончании рабочего дня будет исполняться программный блок, составляющий отчёт о прошедшем дне с Базой данной всех студентов на всех парах.

# **Основания для разработки**

Основанием для разработки является личная инициатива членов команды при выборе темы в рамках дисциплины «Творческий проект» ТПУ.

# **Назначение разработки**

Программа будет использоваться в вузе тремя группами пользователей:

* 1. Преподаватель
  2. Студент
  3. Директор(при получении отчёта посещаемости).

## **3.1. Функциональное назначение**

1. Для преподавателя вуза программа предоставляет возможность выдать студентам сгенерированный QR-код и просмотра присутствующих на паре студентов.
2. Для студентов программа позволяет посредством сканирования QR-кода зарегистрировать присутствие на паре и просмотреть свой статус посещения.
3. Для директора программа позволяет в любой момент получить полную отчётность посещения всех студентов, определённой группы, определённого студента. Так же позволяет получить список студентов, пропустивших наибольшее количество пар.

## **3.2 Эксплуатационное назначение**

Программа должна эксплуатироваться в аудитории. Исполнение программы происходит на сервере. Запущенная с правами студента, она может отображать факт присутствия на паре одного пользователя. Запущенная с правами преподавателя/директора программа может отображать базу данных посещаемости студентов.

# **4 Требования к программе или программному изделию**

## **4.1 Требования к функциональным характеристикам**

### **4.1.1 Требования к составу выполняемых функций**

После запуска программы пользователю отображается форма ввода логина и пароля, показанная на рисунке 1.

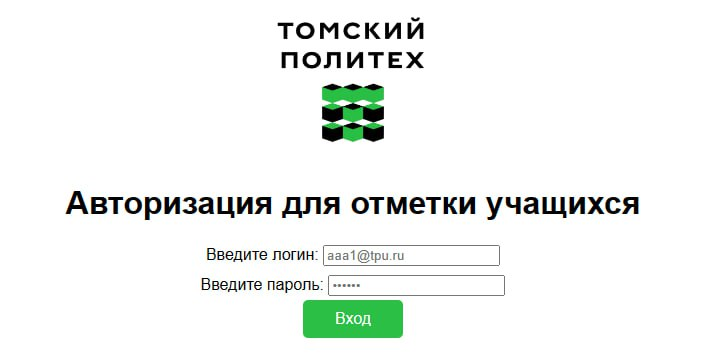


РИС.1

В системе существует всего 2 пользователя — студент и преподаватель/директор. Программа проверяет тип пользователя и открывает соответствующий интерфейс.

Для студентов программа предоставляет следующие возможности:

* Отметить себя на паре
* Удостовериться в собственном присутствии на паре

Для преподавателей программа предоставляет следующие возможности:

* Сгенерировать QR-код на определённую пару
* Вручную добавить студента в список посещения, в случае невозможности сделать это автоматически.

Продолжение и редактирование ТЗ во втором семестре...

# целевая аудитория:

1. Студенты
   1. Отметка присутствия каждого студента традиционным способом может отнимать драгоценное время пары.
   2. В случае, если староста отмечает всех присутствующих (а не преподаватель), то он тратит ещё большее количество времени.
   3. Исключения случаев ошибки проставления присутствия студента, по причине человеческого фактора.
2. Преподаватели
   1. Отметка присутствия каждого студента традиционным способом может отнимать драгоценное время пары, следовательно потеря информации, которую преподаватель мог дать студенту.
   2. Исключения случаев ошибки проставления присутствия студента, по причине человеческого фактора.
   3. Удобство и возможность после пары проставить все пропуски, без присутствия студентов.
3. Директор
   1. Возможность быстро и в любое время получить готовый отчёт, со списком «прогульщиков» для дальнейшего выяснения причин.

# Обзор аналогов:

**Высшие учебные заведения**

В России многие университеты используют личные карточки для учета посещаемости студентов. Например, Московский государственный университет имени М. В. Ломоносова (МГУ) использует личные карточки для доступа в библиотеку, столовую, спортивный зал и другие здания на территории университета. Однако на занятиях до сих пор действует старая система проверки посещаемости.   
Из Положения о текущем контроле успеваемости студентов МГУ: «Учет посещаемости...осуществляется старостами групп путем заполнения под контролем преподавателей журналов посещаемости».

В ТПУ, к примеру, учет посещаемости ведется преподавателями путем внесения данных в модуль «Электронный журнал преподавателям».

Плюсы данного способа ведения учета:

* Отсутствие растрат на дополнительное оборудование
* Студентам не приходится быть вовлеченным в процесс отметки, так как все обязанности лежат на преподавателях и старостах.

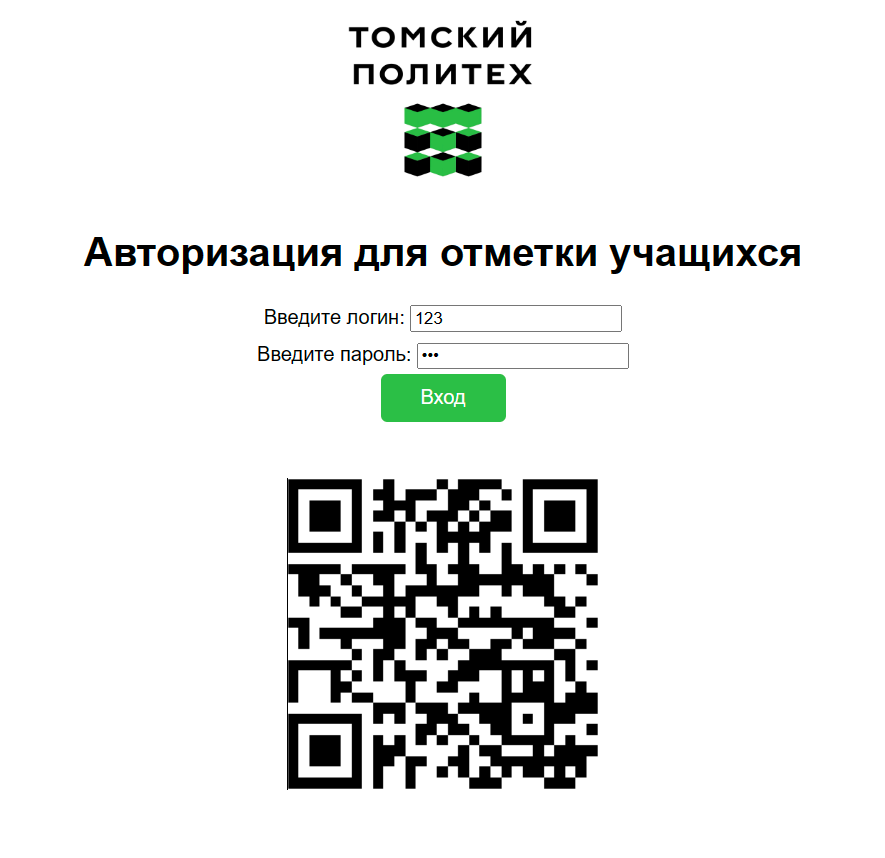
Минусы данного способа ведения учета:

* Занимает большой объем времени от занятия
* Преподаватель может не отмечать студентов

# Обзор интерфейса:

***ПРЕПОДАВАТЕЛЬ:***

Интерфейс сайта для преподавателей состоит из двух полей ввода, куда они должны вводить свои данные чтоб авторизироваться, после авторизации появляется qr-код с ссылкой для перехода на сайт авторизации студентов



**3**

**4**

**2**

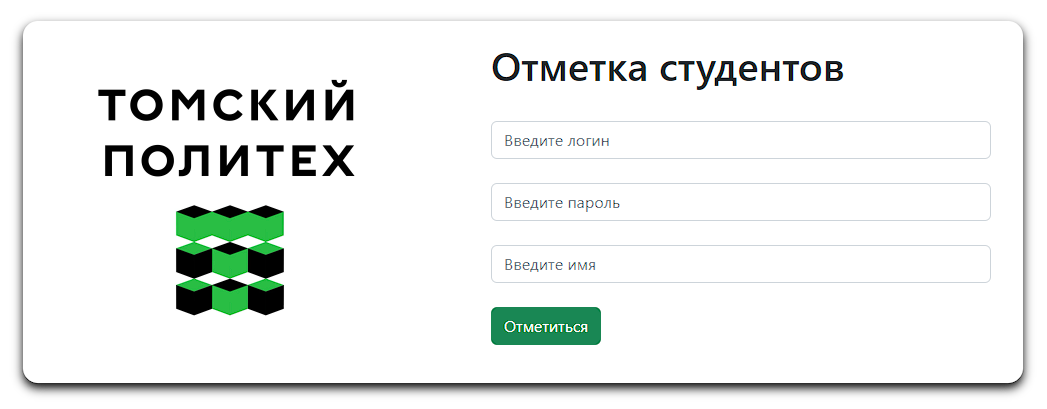
**1**

1. Поле ввода логина для авторизации
2. Поле ввода пароля для авторизации
3. Кнопка входа
4. QR-код выводимый после авторизации

***СТУДЕНТ:***

**2**

**1**



3

**4**

**3**

1) Логотип ТПУ

2) Оглавление страницы

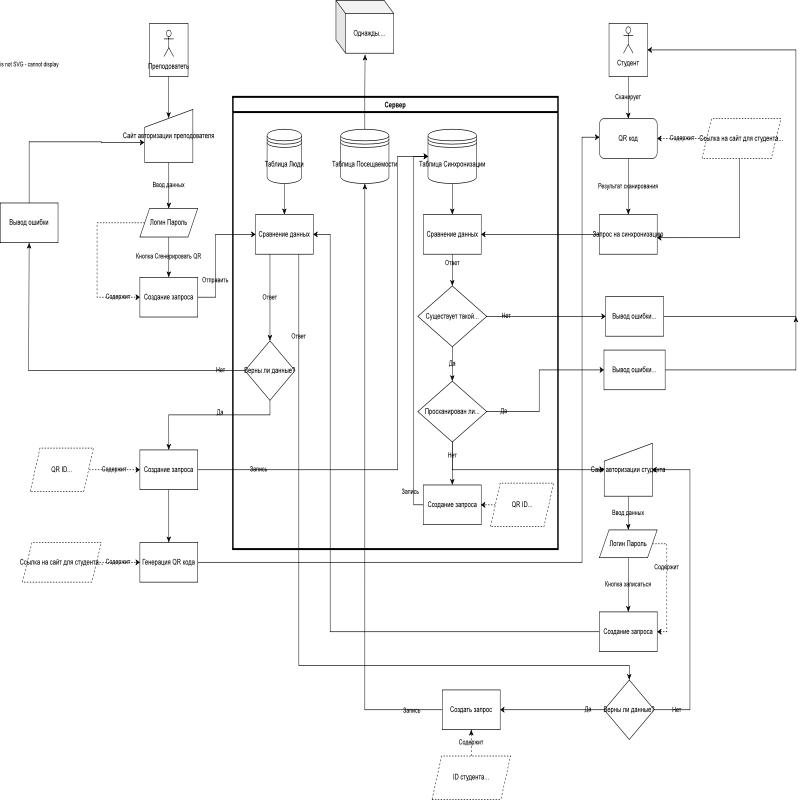
3) Поля ввода данных студента

4) Кнопка для отметки

# UML-диаграмма проекта:

Диаграмма отображает функционирование системы, описывает динамическое поведение веб - ресурса, по сути является алгоритмом работы.

На данной диаграмме изображены взаимодействие двух актёров: Преподавателя и Студента.



UML-диаграмма

*Полная диаграмма -*<https://drive.google.com/file/d/1JDqNkXABd9VBSvPY_qbAOxjsztuD_cHx/view?usp=sharing>

# Реализация front-end части

***ПРЕПОДАВАТЕЛЬ:***

* Код (написанный на html) в [приложении 2.1](#_ПРИЛОЖЕНИЕ 2.) позволяет преподавателю авторизироваться и вывести QR-код который содержит в себе ссылку на сайт авторизации студентов.
* Код (написанный на php) в [приложении 2.2](#_ПРИЛОЖЕНИЕ 2.2.) соединяет сайт и БД для полноценной авторизации
* Код (написанный на js) в [приложении 2.3](#_ПРИЛОЖЕНИЕ 2.3.) создает и выводит QR-код на сайт, на данный момент выводит только один сайт так как не соединили сайт для преподавателя и сайт для студентов.
* Код (написанный на css) в [приложении 2.4](#_ПРИЛОЖЕНИЕ 2.4.) устанавливает максимальный/минимальный размер элементов, цвет, стиль, прозрачность, плавность вывода QR-кода и другие косметические элементы.

***СТУДЕНТ:***

* Код (написанный на HTML) в [приложении 3.1](#_ПРИЛОЖЕНИЕ 3.1.) создает окно авторизации для студентов, и позволяет ввести свои данные для отметки на занятии
* Код (написанный на PHP) в [приложении 3.2](#_ПРИЛОЖЕНИЕ 3.2.) позволяет взаимодействовать с базой данных для того, чтобы вносить изменения в таблицы и передавать текущее состояние авторизации другим членам команды
* Код (написанный на CSS) в [приложении 3.3](#_ПРИЛОЖЕНИЕ 3.3.) устанавливает размеры логотипа и формы, определяет косметические приукрашивания

# Реализация back-end части

* Код (написанный на С++) в [приложении 1](#_ПРИЛОЖЕНИЕ 1.) позволяет проверить присутствии студента на паре, путем сопоставления входных данных студента из QR-кода и Базы данных прошедших через турникет людей (код для Базы Данных в CSV формате).
* Так же сохранить обработанные данные в JSON формате, для дальнейшего составления отчёта и помещения в SQL DB.

Разбор с++ кода с пояснением всех меотдов планируется во 2 семестре...

# Реализация работы с СУБД

MySQL — свободная реляционная система управления базами данных. Является самой популярной из всех серверных БД. Поэтому на просторах интернета достаточно учебников и ресурсов по этой системе. Преимуществами данной системы являются:

1. Простота: в установке, в работе, так как имеет дружелюбный интерфейс и уже предустановленные стандартные настройки,
2. Много функций: предоставляет большую часть функционала SQL,
3. Безопасность: много встроенных функций безопасности от пароля до запрета использование команд удаления,
4. Мощность и масштабируемость: понадобится в дальнейшем,
5. Скорость.

Реализация работы выполнена с помощью Python. Выбор данного языка обусловлен его простотой и быстротой написания кода. Так же, есть удобная библиотека для Python, которая позволяет создавать приложения для СУБД. Данный код из [приложения 4](#_ПРИЛОЖЕНИЕ 4.) реализует соединение с локальной базой данных. Так же реализованы функции с основными командами SQL, такими как: создание таблицы, создание записи в таблицы, отчистка таблицы и обработка данных в JSON формате. Код выполняет запись о присутствии студента в таблицу, доставая необходимые данные о нем из JSON файла.

# Выводы, планы на следующий семестр

Работа была проделана согласно плану.

В будущих семестрах:

* Дописать и отредактировать ТЗ
* Дописать и разобрать C++ код подробнее
* Доделать связь между блоками программы (Python, C++, HTML\CSS\JS\PHP)
* Провести анкетирование среди студентов/преподавателей
* Запустить программу на локальную группу (для сбора информации и ошибок)
* Действия по защите программы на сервере от взлома

# Источники информации

* Metanit.com [ Электронный ресурс ]: Руководство по MySQL. - Режим доступа: <https://metanit.com/sql/mysql/> свободный. (Дата обращения: 26.05.2023)
* Tproger.ru [Электронный ресурс]: Сравнение СУБД. -Режим доступа: <https://tproger.ru/translations/sqlite-mysql-postgresql-comparison/> свободный. (Дата обращение - 25.05.2023)
* Mysql.com [Электронный ресурс]: Документация для СУБД. - Режим доступа: <https://dev.mysql.com/doc/> свободный. (Дата обращения 25.05.2023)
* Mysql.com [Электронный ресурс]: Документация для СУБД совместно с Python. - Режим доступа: <https://dev.mysql.com/doc/connector-python/en/> свободный. (Дата обращения 25.05.2023)
* Mysql.com [Электронный ресурс]: Документация по работе с программой СУБД. - Режим доступа: <https://dev.mysql.com/doc/workbench/en/> (Дата обращения 25.05.2023)
* Статья по С++ «Чтение CSV с помощью С++» - <https://dvsemenov.ru/chtenie-csv-s-pomoshhyu-c/>

# Статья по С++ «Отображение данных в формате json на структуру C++» - <https://habr.com/ru/articles/506506/> (Дата обращения: 13.04.2023)

* Видео «CSS Flexbox. Полный курс» - <https://www.youtube.com/watch?v=XXlw7TUxRVY> (Дата обращения: 26.05.2023)
* Видео «Система регистрации и авторизации на PHP и MySQL базы данных» - <https://www.youtube.com/watch?v=3UtB4QS6EAg&list=PLn3etOLjRRJrHtXpBEJq0AMLdvV3lmXK1&index=6&t=2415s> (Дата обращения: 12.05.2023)
* Видео «Responsive Website Using HTML and CSS Step By Step» - <https://www.youtube.com/watch?v=IXucQAEkIMo&list=PLn3etOLjRRJrHtXpBEJq0AMLdvV3lmXK1&index=5&t=11s> (Дата обращения: 12.05.2023)
* Видео «HTML для начинающих №5. Создание многостраничного сайта. Практический урок» - <https://www.youtube.com/watch?v=lFtUazyLvds&list=PLn3etOLjRRJrHtXpBEJq0AMLdvV3lmXK1&index=3> (Дата обращения: 26.05.2023)
* Положение о текущем контроле успеваемости студентов - <https://sev.msu.ru/wp-content/uploads/2016/10/Polozhenie-o-tekushhem-kontrole-uspevaemosti-studentov-MGU.pdf>
* Приказ ТПУ об «Учете посещаемости студентов» - [https://portal.tpu.ru/standard/attestation/Tab/78\_od\_17\_12\_2012.pdf](https:/portal.tpu.ru/standard/attestation/Tab/78_od_17_12_2012.pdf)
* Видео с изучением основ работы с HTML, PHP и базами данных -
  + <https://www.youtube.com/watch?v=GDVWdYZLM7w&ab_channel=AreaWeb>
  + <https://www.youtube.com/watch?v=3UtB4QS6EAg&t=1986s&ab_channel=%D0%93%D0%BE%D1%88%D0%B0%D0%94%D1%83%D0%B4%D0%B0%D1%80%D1%8C>
  + <https://www.youtube.com/watch?v=ZKU7-ktaa2o&ab_channel=%D0%A5%D0%B0%D1%83%D0%B4%D0%B8%D0%A5%D0%BE%E2%84%A2-%D0%9F%D1%80%D0%BE%D1%81%D1%82%D0%BE%D0%BE%D0%BC%D0%B8%D1%80%D0%B5IT%21>

# Приложение 1.

#define \_CRT\_SECURE\_NO\_WARNINGS

#include <iostream>

#include <string>

#include <fstream>

#include <sstream>

#include <vector>

#include <windows.h>

#include <cstdlib>

#include <ctime>

using namespace std;

int ZAPYAT = 0;

struct StudentRecord {

public:

StudentRecord(

string табель,

string фио,

string отдел,

string должность,

string примечание,

string время\_прохода,

string направление,

string событие,

string объект\_прохода

)

{

ТАБЕЛЬ = табель;

ФИО = фио;

ОТДЕЛ = отдел;

ДОЛЖНОСТЬ = должность;

ПРИМЕЧАНИЕ = примечание;

ВРЕМЯ\_ПРОХОДА = время\_прохода;

НАПРАВЛЕНИЕ = направление;

СОБЫТИЕ = событие;

ОБЪЕКТ\_ПРОХОДА = объект\_прохода;

}

void display() {

cout << " ID: " << ТАБЕЛЬ << endl;

cout << " ФИО: " << ФИО << endl;

cout << " Отдел: " << ОТДЕЛ << endl;

cout << " Должность: " << ДОЛЖНОСТЬ << endl;

cout << "примечание(комната): " << ПРИМЕЧАНИЕ << endl;

cout << " Время прохода: " << ВРЕМЯ\_ПРОХОДА << endl;

cout << " Направление: " << НАПРАВЛЕНИЕ << endl;

cout << " Сбытие(да/нет): " << СОБЫТИЕ << endl;

cout << " Где: " << ОБЪЕКТ\_ПРОХОДА << endl;

cout << endl;

}

string ТАБЕЛЬ;

string ФИО;

string ОТДЕЛ;

string ДОЛЖНОСТЬ;

string ПРИМЕЧАНИЕ;

string ВРЕМЯ\_ПРОХОДА;

string НАПРАВЛЕНИЕ;

string СОБЫТИЕ;

string ОБЪЕКТ\_ПРОХОДА;

};

void displayStudents(vector<StudentRecord>& students) {

for (auto student : students) {

student.display();

}

}

void displayStudentsOnly(vector<StudentRecord>& students,string needly) {

for (auto student : students) {

if(student.ФИО==needly) student.display();

}

}

bool checker(vector<StudentRecord>& students, string place, string name, string path) {

srand((unsigned)time(NULL));

setlocale(LC\_ALL, "ru");

for (auto student : students) {

if (name == student.ФИО) {

ofstream fout;

fout.open(path, ofstream::app);

if (ZAPYAT == 0) {

fout << "\n\t{\n\t \"class\_id\":" << rand() % 400 << ',' << endl

<< " \t \"student\_id\":" << student.ТАБЕЛЬ << ',' << endl

<< " \t \"qr\_check\":" << 1 << ',' << endl;

ZAPYAT = 1;

}

else {

fout << "\n\t,{\n\t \"class\_id\":" << rand() % 400 << ',' << endl

<< " \t \"student\_id\":" << student.ТАБЕЛЬ << ',' << endl

<< " \t \"qr\_check\":" << 1 << ',' << endl;

}

if (place == student.ОБЪЕКТ\_ПРОХОДА && student.НАПРАВЛЕНИЕ == "Вход") {

cout << "\nНА МЕСТЕ";

fout << "\t \"skud\_check\":" << 1 << endl << "\t}";

fout.close();

return 1;

}

else {

cout << "\nНЕТУ";

fout << "\t \"skud\_check\":" << 0 << endl << "\t}";

fout.close();

return 0;

}

}

}

}

void displayStudentsOnlyStuding(vector<StudentRecord>& students,string needly) {

for (auto student : students) {

if ((student.ОБЪЕКТ\_ПРОХОДА[0] == 'У'

|| student.ОБЪЕКТ\_ПРОХОДА[0] == 'Н'

|| student.ОБЪЕКТ\_ПРОХОДА[0] == 'К') && student.НАПРАВЛЕНИЕ=="Вход" && student.ФИО == needly)

student.display();

}

}

string ТАБЕЛЬ;

string ФИО;

string ОТДЕЛ;

string ДОЛЖНОСТЬ;

string ПРИМЕЧАНИЕ;

string ВРЕМЯ\_ПРОХОДА;

string НАПРАВЛЕНИЕ;

string СОБЫТИЕ;

string ОБЪЕКТ\_ПРОХОДА;

string tempString;

vector<StudentRecord> students;

int main()

{

time\_t now = time(0);

srand((unsigned)time(NULL));

setlocale(LC\_ALL, "Russian");

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

const time\_t tm = time(nullptr);

char buf[64];

strftime(buf, std::size(buf), "%d.%m.%Y", localtime(&tm));

ifstream inputFile;

inputFile.open("C:\\temp\\NOVEMBER.csv");

string line = "";

string path = "C:\\bufer\\";

path += buf;

path += ".txt";

ofstream fout;

fout.open(path);

fout << "[";

fout.close();

while (getline(inputFile, line) && ВРЕМЯ\_ПРОХОДА[0] != '2' && ВРЕМЯ\_ПРОХОДА[0] != '9') {

stringstream inputString(line);

getline(inputString, ТАБЕЛЬ, ';');

getline(inputString, ФИО, ';');

getline(inputString, ОТДЕЛ, ';');

getline(inputString, ДОЛЖНОСТЬ, ';');

getline(inputString, ПРИМЕЧАНИЕ, ';');

getline(inputString, ВРЕМЯ\_ПРОХОДА, ';');

getline(inputString, НАПРАВЛЕНИЕ, ';');

getline(inputString, СОБЫТИЕ, ';');

getline(inputString, ОБЪЕКТ\_ПРОХОДА, ';');

// if (ФИО == "Будзиаловский Павел Борисович") {

StudentRecord student(ТАБЕЛЬ, ФИО, ОТДЕЛ, ДОЛЖНОСТЬ, ПРИМЕЧАНИЕ, ВРЕМЯ\_ПРОХОДА, НАПРАВЛЕНИЕ, СОБЫТИЕ, ОБЪЕКТ\_ПРОХОДА);

students.push\_back(student);

line = "";

}

string Who;

string Where;

while (1) {

cout << "Кого проверить...";

getline(cin, Who);

if (Who == "1") break;

cout << "\nКуда должен был зайти студент?";

getline(cin, Where);

checker(students, Where, Who, path);

cout << endl;

}

fout.open(path, ofstream::app);

fout << "\n]";

fout.close();

//displayStudentsOnly(students, Who);

}

# Приложение 2.1.

<!doctype html>  
<html lang="ru">  
<head>  
 <link rel="stylesheet" href="style.css">  
 <meta charset="UTF-8">  
 <meta name="viewport"  
 content="width=device-width, user-scalable=no, initial-scale=1.0, maximum-scale=1.0, minimum-scale=1.0">  
 <meta http-equiv="X-UA-Compatible" content="ie=edge">  
 <title> Отметка студентов </title>  
</head>  
<body>  
 <img src="https://img.artlebedev.ru/tpu/tpu-logo.png" height="170" width="200" alt="Логотип ТПУ" align="top"/>  
 <form>  
<div>  
 <h1> Авторизация для отметки учащихся </h1>  
 <label>  
 Введите логин:  
 <input type="text" name="login" spellcheck="false" placeholder="aaa1@tpu.ru" required>  
 </label>  
</div>  
<div>  
 <label>  
 Введите пароль:  
 <input type="password" name="password" spellcheck="false" placeholder="••••••" required>  
 </label>  
</div>  
 <div class="wrapper">  
 <div class="form">  
 <button class= "c-button" type="button">Вход</button>  
 </div>  
 <div class="qr-code">  
 <img src="" alt="qr-code">  
 </div>  
 </div>  
 <script src="automqr.js"></script>  
 </form>  
</body>  
</html>

# Приложение 2.2.

<?php  
$login = filter\_var(trim($\_POST['login']),  
 *FILTER\_SANITIZE\_STRING*);  
$pass = filter\_var(trim($\_POST['pass']),  
 *FILTER\_SANITIZE\_STRING*);  
  
$mysql = new mysqli('localhost', 'root', 'root', 'register-bd');  
  
$result = $mysql->query("SELECT \* FROM 'users' WHERE 'login' = '$login' AND 'pass' = '$pass'");  
$user = $result->fetch\_assoc();  
if(count($user) == 0) {  
 echo "Пользователь не найден";  
 exit();  
}  
  
$mysql->close();  
header('Location: /');  
?>

# Приложение 2.3.

const *wrapper* = *document*.querySelector(".wrapper"),  
 *generateBtn* = *wrapper*.querySelector(".form button"),  
 *qrImg* = *wrapper*.querySelector(".qr-code img");  
let *preValue*;  
  
*generateBtn*.addEventListener("click", () => {  
 *qrImg*.src = `https://api.qrserver.com/v1/create-qr-code/?size=250x250&data=https://steamcommunity.com/id/lilawert/`;  
 *qrImg*.addEventListener("load", () => {  
 *wrapper*.classList.add("active");  
 });  
});  
  
*qrInput*.addEventListener("keyup", () => {  
 if(!*qrInput*.value.trim()) {  
 *wrapper*.classList.remove("active");  
 *preValue* = "";  
 }  
});

# Приложение 2.4.

body {  
 font-family: Arial, "Helvetica Neue", Helvetica, sans-serif;  
 text-align: center;  
 line-height: 30px;  
}  
.c-button {  
 min-width: 100px;  
 font-family: inherit;  
 appearance: none;  
 border: 0;  
 border-radius: 5px;  
 background: #29bf46;  
 color: #fff;  
 padding: 10px 25px;  
 font-size: 1rem;  
 cursor: pointer;  
}  
.c-button:hover {  
 background: #26963a;  
}  
.c-button:focus {  
 outline: none;  
 box-shadow: 0 0 0 4px #cbeecc;  
}  
.qr-code {  
 opacity: 0;  
 display: flex;  
 padding: 45px 0;  
 border-radius: 5px;  
 align-items: center;  
 pointer-events: none;  
 justify-content: center;  
}  
.wrapper.active .qr-code {  
 opacity: 1;  
 pointer-events: auto;  
 transition: opacity 0.5s 0.05s ease;  
}  
.qr-code img {  
 width: 250px;  
}

# Приложение 3.1.

<!DOCTYPE html>

<html>

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<meta http-equiv="X-UA-Compatible" content="ie-edge">

<title>Форма регистрации</title>

<link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.3/dist/css/bootstrap.min.css" integrity="sha384-rbsA2VBKQhggwzxH7pPCaAqO46MgnOM80zW1RWuH61DGLwZJEdK2Kadq2F9CUG65" crossorigin="anonymous">

<link rel="stylesheet" href="style.css">

</head>

<body>

<div class="container mt-4">

<div class="row">

<div class="col">

<img class="logo" src="img/tpu-logo.png" alt="TPU LOGO">

</div>

<div class="col">

<h1>Отметка студентов</h1> <br>

<form action="check.php" method="post">

<input type="text" class="form-control" name="login" id="login" placeholder="Введите логин" required> <br>

<input type="password" class="form-control" name="pass" id="pass" placeholder="Введите пароль" required> <br>

<input type="text" class="form-control" name="name" id="name" placeholder="Введите имя" required><br>

<button class="btn btn-success" type="submit">Отметиться</button><br>

</form>

</div>

</div>

</div>

</body>

</html>

# Приложение 3.2.

<?php

$login = filter\_var(trim($\_POST['login']), FILTER\_SANITIZE\_STRING);

$pass = filter\_var(trim($\_POST['pass']), FILTER\_SANITIZE\_STRING);

$name = filter\_var(trim($\_POST['name']), FILTER\_SANITIZE\_STRING);

if (mb\_strlen($login) < 5 || mb\_strlen($login) > 90) {

echo "переделай сука";

exit();

}

$mysql = new mysqli('localhost', 'root', 'root', 'tpu-authorization-test');

$result = $mysql->query('INSERT INTO `dlya-sashi` (id-student, time)

VALUES($name, NOW())');

$result = $mysql->query('INSERT INTO `users` (check, time)

VALUES(true, NOW())');

$mysql->close();

?>

# Приложение 3.3.

.logo{

width: 350px;

height: 322px;

border-radius: 10%;

}

form{

width: 500px;

}

.row{

width: 1000px;

margin: 0 auto;

padding: 20px;

box-shadow: 0 5px 10px #000000;

border-radius: 15px;

}

# Приложение 4.

# Подключение необходымых данных для подключения к СУБД

from config import CONFIG, DB\_NAME

import mysql.connector

from mysql.connector import errorcode

# Подключение библиотеки для работы с JSON файлами

import json

# Функция создания таблицы

def create\_table(cur, table):

try:

cur.execute(table)

except mysql.connector.Error as err:

if err.errno == errorcode.ER\_TABLE\_EXISTS\_ERROR:

print("Table already exist")

else:

print("Something went wrong (create table): {}".format(err.msg))

else:

print("Table created")

# Функция добавления записи в таблицу успеваемости

def add\_check(cur, data, table):

query = ('INSERT INTO {table} '

'(class\_id, student\_id, qr\_check, skud\_check) '

'VALUES (%(class\_id)s, %(student\_id)s, %(qr\_check)s, %(skud\_check)s)'.format(table=table))

try:

cur.execute(query, data)

except mysql.connector.Error as err:

print("Something went wrong (add\_check): {}".format(err))

# Функция отчистки таблицы

def clear\_table(cur, table):

try:

cur.execute("DELETE FROM "+ table)

except mysql.connector.Error as err:

print("Something went wrong (clear\_table): {}".format(err))

# Функция обработки JSON файла

def json\_to\_dict(filename = "data.json"):

try:

file = open(filename,"r",encoding="utf-8")

data = json.loads(file.read(), object\_hook = dict)

return data

except OSError as err:

print("There is problem in json to db func: {}".format(err))

def txt\_to\_dict(filename = "text.txt"):

try:

file = open(filename)

data = eval(file.read())

return data

except BaseException as err:

print("There is a problem in txt\_to\_dict func:", err)

#Соеднинение с базой данных

try:

Connection = mysql.connector.connect(\*\*CONFIG)

Cursor = Connection.cursor()

Cursor.execute(

'USE {}'.format(DB\_NAME)

)

data = txt\_to\_dict("19.06.2023.txt")

for stud in data:

add\_check(Cursor,stud,'attendance')

# Обработка ошибок

except mysql.connector.Error as err:

if err.errno == errorcode.ER\_ACCESS\_DENIED\_ERROR:

print("Something went wrong with your login or password")

elif err.errno == errorcode.ER\_BAD\_DB\_ERROR:

print("This database does not exist")

else:

print(err)

else:

try:

Connection.commit()

Cursor.close()

Connection.close()

except mysql.connector.Error as err:

print("Something went wrong: {}".format(err))

else: print("\nThere are not problems with commiting and closing connections\n")